What is claimed is:

1. An NVRAM fail-over controller comprising:

An NVRAM device connected to a host computer, the host computer having the ability to directly control the NVRAM device;

An embedded processor on the NVRAM fail-over controller that is powered by back-up power;

A network interface on the NVRAM fail-over controller that is powered by back-up power.

2. A method of using a controller of claim 1, the method comprising of the controller performing the following steps:

the controller determining or being told that the host computer has failed;

transmitting NVRAM data to another computer.

3. A method of using a controller of claim 1, the method comprising of the controller performing the following steps:

the controller determining or being told that the host computer has failed;

responding to requests from another computer to transmit part or all of the NVRAM data.

4. A duplicity of controllers of claim 1 connected to each member of a cluster of host computers and connected to each other by network interconnections, thereby enabling fail-over operations between cluster members.

5. A controller of claim 1 using non-transparent bus bridges.

6. A method of using a controller of claim 5 in which the bridges act as firewalls to protect one portion of the system from failures on the other side of the firewall, the method comprising of the controller performing the following steps:

the controller determining or being told that the host computer has failed;

the controller programming its bridge between itself and the host computer to not forward requests through the bridge;

the controller continuing its operations including transmission or storage of NVRAM data;

the controller receiving a network message that it is OK to reestablish a connection through the bridge or the host computer proving itself healthy enough to reprogram the bridge.

7. A method of using a controller of claim 5 in which the bridges act as firewalls to protect one portion of the system from failures on the other side of the firewall, the method comprising of the host computer performing the following steps:

the host computer determining or being told that the controller has failed;

the host computer programming its bridge between itself and the controller to not forward requests through the bridge;

the host computer continuing its operations without storing data to or reading data from NVRAM, or continuing to use NVRAM but not being able to use fail-over capabilities of the controller;

the host computer determining or being told that it is OK to reestablish a connection through the bridge or the controller proving itself healthy enough to reprogram the bridge.

8. A controller of claim 1 with the ability to add daughter cards.

9. A controller of claim 8 in which the daughter card is a network controller under control of the host computer.

10. A controller of claim 8 in which the daughter card is a disk or RAID controller under control of the host computer.

11. A controller of claim 8 in which the daughter card is a disk or RAID controller under control of the embedded processor.

12. A controller of claims 10 or 11 used as a RAID device with NVRAM under control of the host computer.

13. A controller of claim 1 used as an NVRAM device that preserves data during long outages by sending it to another host computer over a network or to a disk attached as in claims 10 or 11 and that retrieves such data back into NVRAM at a later time.

14. A controller of claim 1 used as an NVRAM device that preserves data during long outages by sending it to another computer over a network such that the other computer can take over operations from the first host computer, such as managing storage devices and using network addresses from the host computer.

15. A method as in claims 11 or 12 that further keeps the remote server system almost up-to-date during normal operation to reduce the time to bring it fully up-to-date when a failure occurs, such method comprising of the controller performing the following steps:

receiving data in NVRAM from the host computer;
telling the host computer that the transfer is complete;
informing another computer that the data is available;
transfering data to the other computer at a suitable time;
keeping track of what data still remains to be sent;

and at a later time

the controller determining or being told that the host computer has failed;
transfering all unsent data to the other computer.

16. A controller of claim 1 with a watchdog timer to reset the embedded processor.

17. A controller of claim 1 with the ability for the host computer to reset the embedded processor.